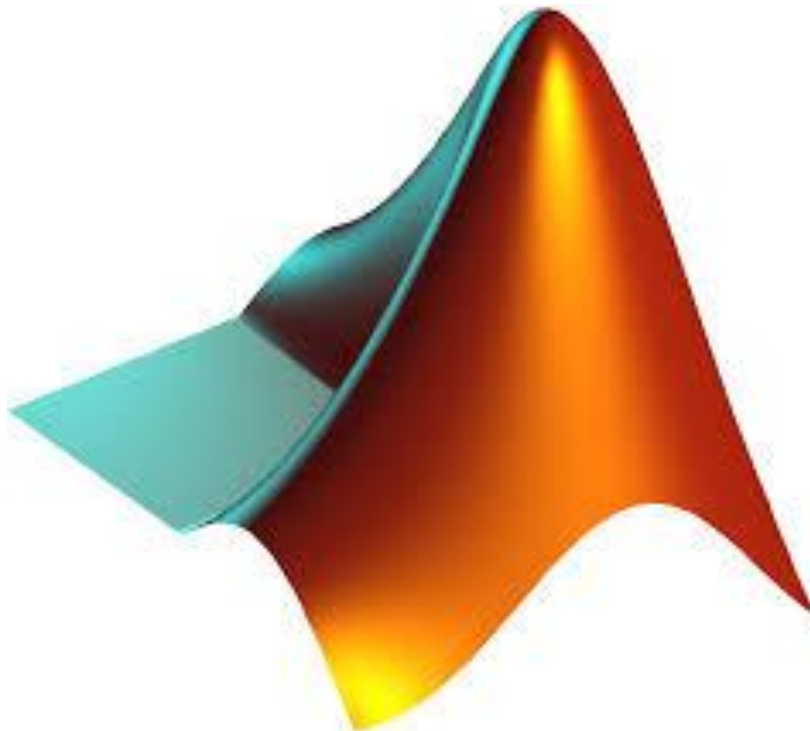


# Traitement de l'image

## RAPPORT DU TP2



### Réalisé par :

- DAHOU Yassine
- DANNOUNE Sara
- ELOUAZZANI Kenza
- SOUKRAT Yahia

### Encadré par :

M.EL BOUSTANI Hakim

## ***Objectif du TP :***

L'objectif poursuivi à travers ce TP est l'initiation aux techniques de filtrage d'images avec Matlab. Il s'agit de comparer plusieurs filtres sur plusieurs types de bruits et d'appliquer plusieurs filtres pour détecter les contours dans une image. Le filtre de Butterworth sera aussi étudié.

## ***Filtrage de bruit : filtre médian, filtre gaussien, filtre moyonneur :***

D'abord on va lire l'image avec laquelle on va travailler :



Nous avons utilisé une image de 256x256 Pixels. Ensuite on ajoute plusieurs type de bruits à l'image fournie et on applique à chaque bruit trois type de filtre (médian, gaussien, moyonneur).

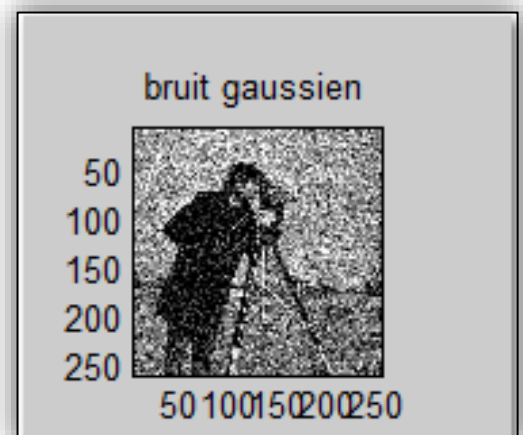
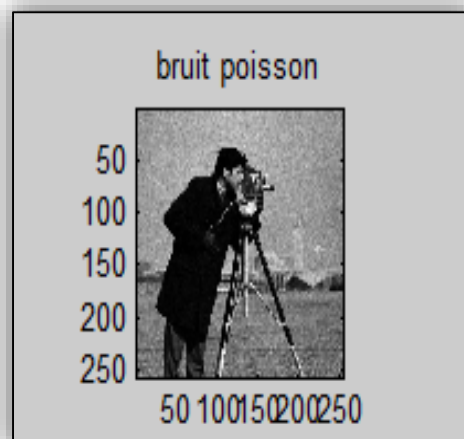
## ➤ Ajout de bruit (poivre et sel, gaussien, poisson) :

```
f=imread('C:\Users\user\Desktop\cameraman.gif','gif');

imagesc(f);colormap(gray);
title('Image de base');
% ajout des bruits
% sel et povre
bsp = imnoise(f,'salt & pepper', 0.02);

%gaussien
bg = imnoise(f,'gaussian',0,0.05);

%poisson
bp = imnoise(f,'poisson');
```



## ➤ Filtrage des images à l'aide des 3 filtres :

### Code :

```
%filtrage avec le filtre moyenneur
N = ones(3)/9;
mysp = imfilter(bsp,N);
myg = imfilter(bg,N);
myp = imfilter(bp,N);

%filtrage avec le filtre gaussien
fig = fspecial('gaussian',[256,256],5);

gsp = imfilter(bsp,fig);
gg = imfilter(bg,fig);
gp = imfilter(bp,fig);

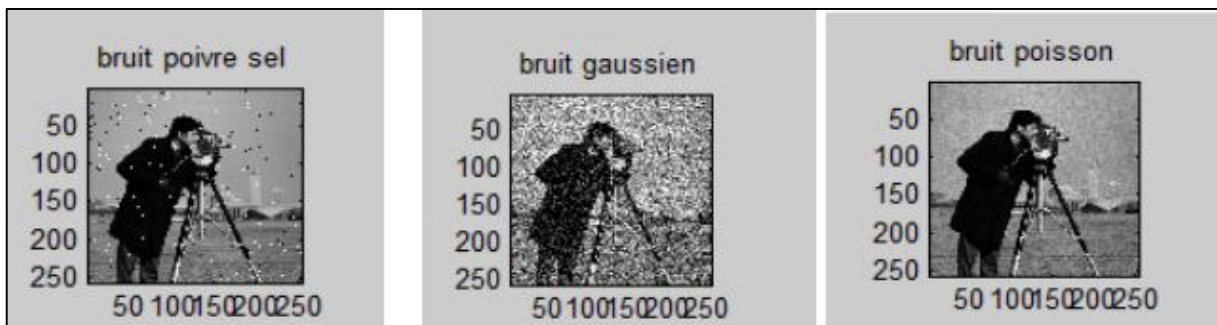
%filtrage avec le filtre median
msp = medfilt2(bsp);
mg = medfilt2(bg);
mp=medfilt2(bp);
```

### Affichage :

```
%Affichage des images

figure, imshow(f);
figure(2)
subplot(4,3,1), subimage(bsp), title('bruit poivre sel')
subplot(4,3,2), subimage(bg), title('bruit gaussien')
subplot(4,3,3), subimage(bp), title('bruit poisson')
subplot(4,3,4), subimage(gsp), title('Ps filtré gaussien')
subplot(4,3,5), subimage(gg), title('bruit gaussien filtré gauss')
subplot(4,3,6), subimage(gp), title('poisson filtré gaussien')
subplot(4,3,7), subimage(mysp), title('ps filtré moyenneur')
subplot(4,3,8), subimage(myg), title('gaussien filtré moyenneur')
subplot(4,3,9), subimage(myp), title('poisson filtré moyenneur')
subplot(4,3,10), subimage(msp), title('ps filtré median')
subplot(4,3,11), subimage(mg), title('gaussien filtré median')
subplot(4,3,12), subimage(mp), title('poisson filtré median')
```

## ✚ Type de bruit :



## ✚ Filtrage par le filtre gaussien :



## ✚ Filtrage par le filtre moyennneur :



## ✚ Filtrage par le filtre median :



## Conclusion :

- Pour filtrer le bruit poivre et sel on utilise un filtre median
- Pour filtrer le bruit poisson on utilise un filtre myennneur ou median

## ➤ Détection de contour :

Pour détecter les contours on applique les filtres de Prewitt, Sobel et Canny sur l'image fournie et on compare les résultats.

## ✚ Code Matlab : Filtre Sobel :

```
img=imread('C:\Users\Saad\Desktop\cameraman.gif');
IF=edge(img,'sobel');
imshow(IF);
figure(1);
I1=imnoise(img,'gaussian');
IF1=edge(I1,'sobel');
imshow(IF1);
figure(3);
```

Image d'origine filtrée

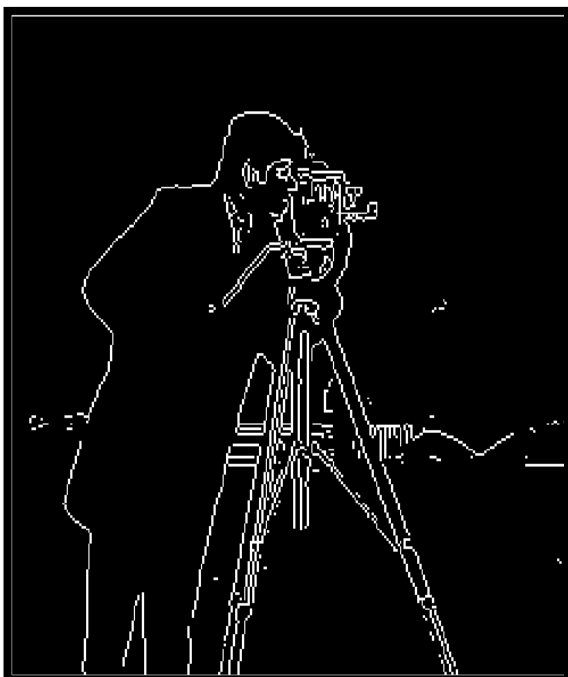


Image bruitée filtrée : Sobel



## Code Matlab : Filtre Prewitt :

```
img=imread('C:\Users\Saad\Desktop\cameraman.gif');
IF1=edge(img,'prewitt');
imshow(IF1);
figure(2);
I1=imnoise(img,'gaussian');
IF1=edge(I1,'prewitt');
imshow(IF1);
figure(3);
```

Image d'origine filtrée

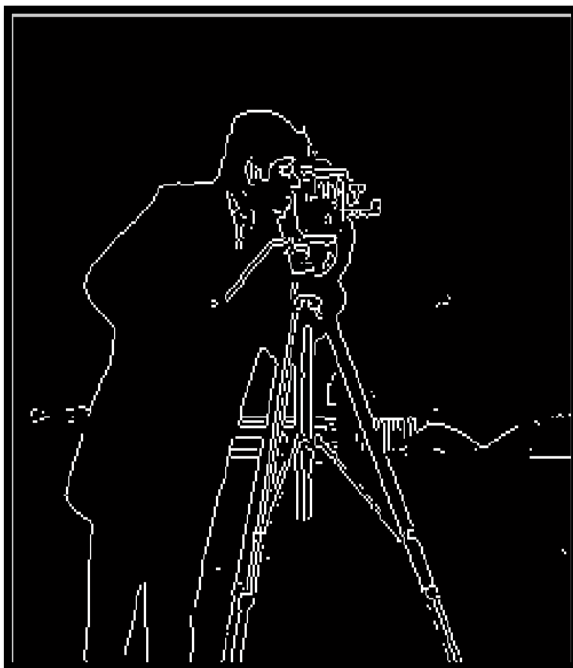


Image bruitée filtrée : Prewitt



## Code Matlab : Filtre canny:

```
img=imread('C:\Users\Saad\Desktop\cameraman.gif');
IF1=edge(img,'canny');
imshow(IF1);
figure(2);
I1=imnoise(img,'gaussian');
IF1=edge(I1,'canny');
imshow(IF1);
figure(3);
```

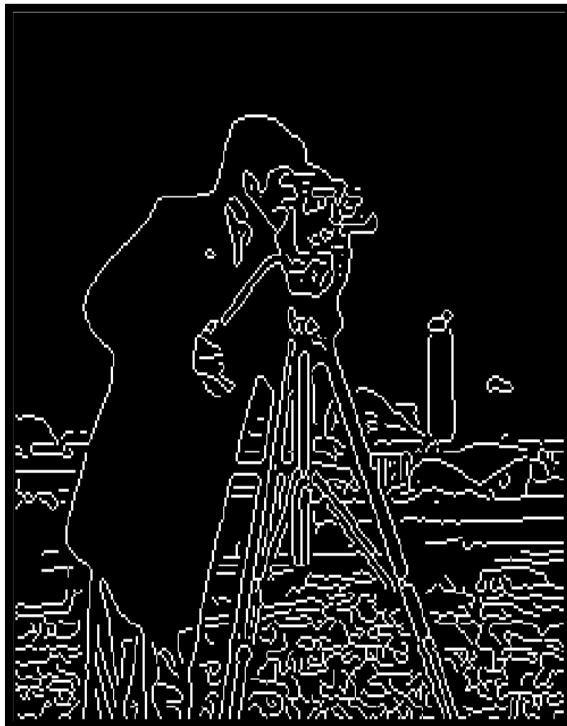


Image d'origine filtrée

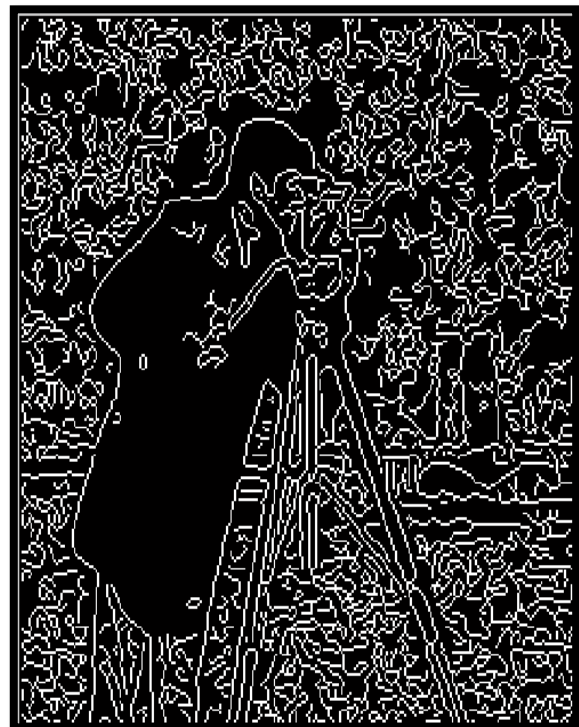


Image bruitée filtrée : Canny



## Filtre ideal pass bas :

### Script :

```
%Filtre passe bas idéal
f=imread('C:\Users\yassine dahou\Pictures\sara','gif');

N = 256;
DO =32;
H = zeros(N,N);
for x = 0: N-1
    for y = 1:N-1
        D = sqrt((x-128)^2 + (y-128)^2);
        if D <= DO
            H(x,y) = 255;
        end
    end
end
figure , imshow(H); title('Le masque du filtrage dans le domaine Frq')
F = fft2(f); figure , imshow(F), title('Image dans le domaine frq')
Fsh = fftshift(F); figure , imshow(Fsh),title('Image centrée dans le D Frq')
|
```

### Affichage :

## Filtre Butterworth pass-bas :

### Script:

```
f=imread('C:\Users\yassine dahou\Pictures\sara','gif');
|
% Filtre Butterworth pass-bas:

D = 32;
n = 1;

[umax vmax] = size(Fsh);
for u = 1: umax
    for v = 1: vmax
        Hb(u,v) = 1/(1+(sqrt(2)-1)*(sqrt(((umax/2 - (u-1)).^2+(vmax/2 - (v-1)).^2)/D).^2)).^(2*n));
    end;
end

figure , imshow(Hb); title('Le masque du filtrage Butterworth PB dans le domaine Frq')
F = fft2(f); figure , imshow(F), title('Image dans le domaine frq')
Fsh = fftshift(F); figure , imshow(Fsh),title('Image centrée dans le D Frq')
Fb1p = Fsh.*Hb; figure , imshow(Fb1p),title('Image filtrée Butterworth PB dans le D Frq')
fb1p = ifft2(Fb1p);
fb1p = sqrt(real(fb1p).^2+imag(fb1p).^2);
imagesc(fb1p), colormap(gray),title('Image filtrée Butterworth PB dans le domaine réel')
```

### Affichage :

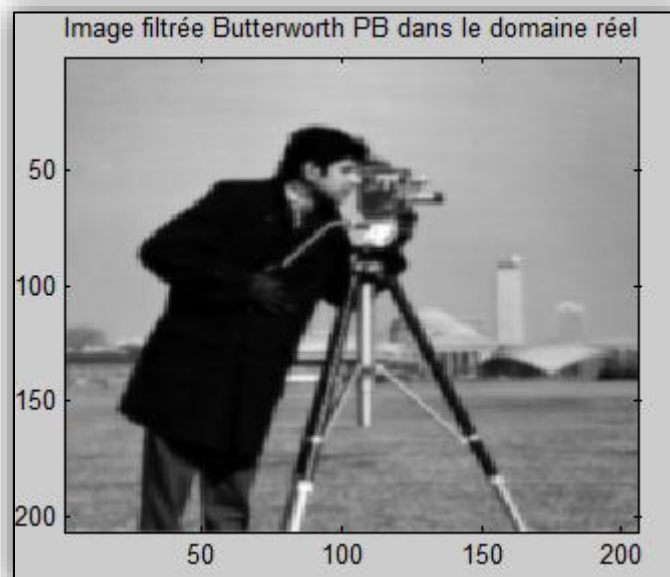


Image centrée dans le D Frq

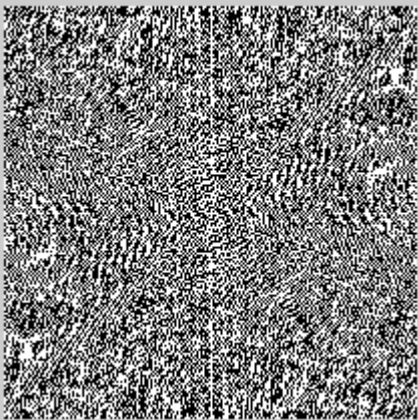
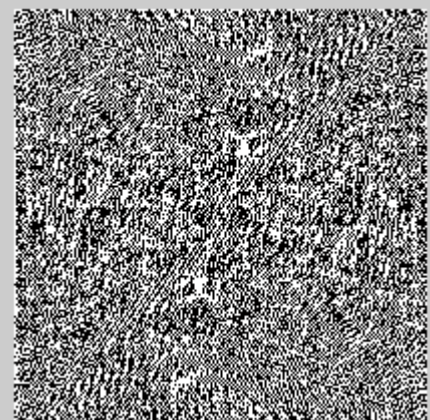


Image dans le domaine frq



Le masque du filtrage Butterworth PB dans le domaine Frq

